



International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2(3) (2026)



ijerst.editor@gmail.com
editor@ijerst.com

Research Paper

A Data-Driven Learning Analytics Architecture for Continuous Student Outcome Prediction and Intelligent Faculty Allocation

S. Krishna Reddy¹, Chilka Sunil Joshi¹, Bogiboina Satish¹, Durgam Ravinder¹, G. Laxmi Narasimha Reddy¹

¹Department of Computer Science and Engineering, ¹Sree Dattha Institute of Engineering and Science, Nagarjuna Sagar Road, Sheriguda, Ibrahimpatnam, Rangareddy Dist, 501510, Telangana, India.

Abstract

The rapid expansion of student enrolment in modern educational institutions has created a strong need for smart and efficient academic management solutions. Conventional systems based on manual record-keeping and paper registers are not only time-consuming but also highly susceptible to human errors, data redundancy, and poor accessibility. These limitations make it difficult to monitor student progress and manage academic workflows effectively. To overcome these issues, this study introduces a Web-based Student Performance Management System developed using Django framework and MySQL. The proposed system provides a centralized and role-based platform where Admins, Faculty, and Students can interact seamlessly. It incorporates essential modules such as Attendance Management, Marks Entry, Assignment Handling, Study Material Sharing, Feedback System, and Internal Communication, thereby improving coordination and reducing manual effort. A key contribution of this system is the integration of Machine Learning (ML) for academic performance prediction. Models such as Naive Bayes Classifier (NBC) and Extreme Gradient Boosting (XGB) are applied and compared. Experimental results indicate that XGB achieves higher prediction accuracy and robustness, making it the optimal model for performance classification. The ML module analyses historical academic data to categorize students into performance levels, enabling early intervention and data-driven decision-making. The SPMS effectively combines Web Technologies, Database Management Systems (DBMS), and ML to enhance efficiency, ensure data accuracy, and provide real-time accessibility, making it a scalable and intelligent solution for modern education systems.

Keywords: Educational Data Mining, Student Performance Management System (SPMS), Role-Based Access Control, Academic Performance Prediction, Web-Based Application

1. Introduction

Student retention and success remain critical concerns in higher education, as many learners require continuous academic, emotional, and financial support to successfully complete their studies. To address these needs, universities have increasingly adopted structured support systems aimed at improving student engagement and persistence. Grounded in Self-Determination Theory (SDT), student motivation is influenced by three essential psychological needs: autonomy (sense of control over actions), competence (ability to perform effectively), and relatedness (connection with others). Institutions that actively foster these dimensions are more effective in enhancing student outcomes and long-term academic success. In this context, Early Alert Systems (EAS) have emerged as a proactive strategy to identify students who may be at academic risk and to facilitate timely interventions, as shown in figure 1. These systems enable faculty and academic advisors to detect early signs of disengagement or poor performance and initiate supportive actions such as counselling, tutoring, or mentoring. According to Vincent Tinto's theory of student departure, meaningful interactions between students and institutional representatives significantly strengthen social integration, thereby reducing dropout

rates [1]. Early alert mechanisms contribute to this integration by promoting timely communication and personalized academic support.

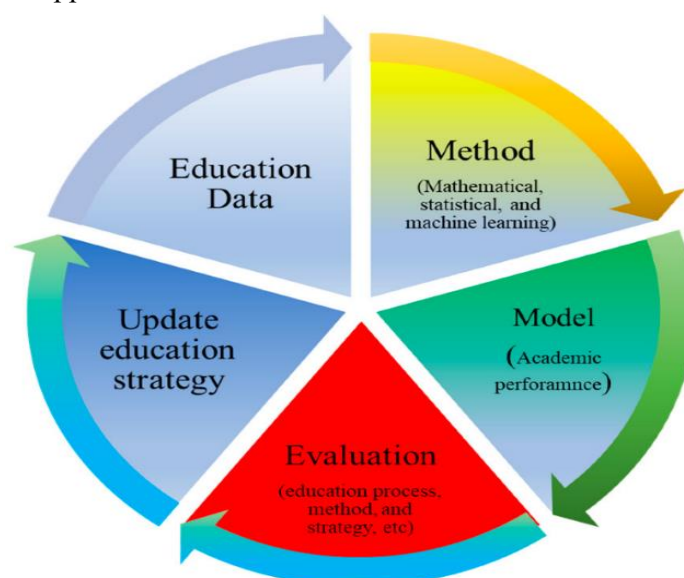


Figure. 1: student academic methods

The adoption of early alert systems is widespread, with studies indicating that approximately 85–90% of higher education institutions in the United States had implemented such systems as early as 2014 [2]. Their effectiveness has also been demonstrated globally across institutions in Europe, Africa, Asia, and Oceania, where they have improved student outcomes through accurate identification of at-risk individuals and better resource allocation [3]. Although these systems primarily function as facilitators rather than direct service providers, they act as critical gateways connecting students to institutional resources. For instance, an alert may encourage a student to consult an academic advisor, who may then recommend additional support services such as tutoring or counselling. The present study builds upon these theoretical foundations by examining student perceptions and responses to an early alert mechanism known as Academic Status Reports (ASRs) [4]. By analysing student awareness and engagement with such systems, this research aims to provide deeper insights into how institutional interventions influence student behaviour and persistence in higher education environments [5].

2. Literature Survey

Alhothali, et al. [6] conducted an extensive and systematic review of contemporary research focusing on the prediction of online learners' outcomes using both Machine Learning (ML) and Deep Learning (DL) techniques. Their study makes a significant contribution by organizing and categorizing the diverse set of features extracted from online learning environments, such as interaction logs, assessment data, and behavioral patterns. Furthermore, the authors analyzed various prediction targets, including grades, completion rates, and engagement levels. They also examined feature engineering and extraction techniques, evaluation metrics like accuracy, precision, recall, and F1-score, and proposed a structured taxonomy for classifying prior research. In addition, the study highlights key challenges such as data heterogeneity, lack of standardization, and limited interpretability of models, providing valuable insights for future research directions.

Alruwais, et al. [7] developed a Machine Learning (ML)-driven framework aimed at continuously evaluating student performance and knowledge progression throughout their academic journey. The system identifies critical factors that significantly influence student success, such as attendance, assignment completion, and participation. A notable aspect of their work is the emphasis on feature selection, where the authors analyze how models perform when trained using only the most influential features. Their findings demonstrate that reducing irrelevant or redundant data can improve model

efficiency while maintaining or even enhancing prediction accuracy. Adefemi, et al. [8] addressed the limitations of traditional prediction approaches by proposing a hybrid Deep Learning (DL) model that integrates Convolutional Neural Networks (CNN) with Bidirectional Gated Recurrent Units (BiGRU). This combination enables the model to capture both spatial and temporal patterns in student data, leading to improved predictive performance. The study also tackles critical preprocessing challenges, including handling missing values, mitigating class imbalance using resampling techniques, and selecting the most relevant features. As a result, the proposed model achieves higher accuracy and robustness compared to conventional ML methods.

Namoun, et al. [9] performed a comprehensive systematic literature review by analyzing 62 relevant studies related to student performance prediction. The authors structured their analysis around three primary dimensions: the types of predicted learning outcomes (such as grades and rankings), the predictive models employed, and the key factors influencing student success. Using established review methodologies such as PICO and PRISMA, they ensured a rigorous and transparent synthesis of findings. Their results indicate that academic performance is commonly measured through grades and class standings, and that factors such as engagement, prior academic history, and demographic attributes play a crucial role in prediction accuracy. Jawad, et al. [10] proposed an approach that focuses on analyzing student engagement patterns and predicting academic performance using a Random Forest (RF) classifier combined with the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance. The study utilized the Open University Learning Analytics Dataset (OULAD), which provides rich information about student interactions over multiple time periods. By constructing detailed student profiles based on assessment scores and engagement metrics, the model effectively captures learning behavior trends and improves prediction reliability.

Baashar, et al. [11] explored the application of Artificial Neural Networks (ANN) in predicting academic performance by reviewing a wide range of studies, particularly in higher education contexts. Their analysis revealed that ANN models are frequently integrated with data mining and statistical analysis techniques to enhance predictive capabilities. The study also highlights that there is no universal standard for selecting input features, as feature selection largely depends on the specific context, dataset characteristics, and research objectives. This variability underscores the importance of domain knowledge in designing effective prediction models. Mohammad, et al. [12] investigated the effectiveness of multiple Machine Learning techniques for early prediction of student performance through two simulation scenarios. The first simulation employed Traditional Machine Learning Classifiers (TMLCs) on benchmark datasets, including Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), Decision Tree (DT), Multi-Layer Perceptron (MLP), Random Forest (RF), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA). Their comparative analysis demonstrated that ensemble and nonlinear models generally outperform simpler classifiers in terms of predictive accuracy and generalization.

Shou, et al. [13] introduced a sophisticated student performance prediction model based on multidimensional time-series data analysis. The model incorporates various data sources, including students' learning behaviors, assessment results, and demographic information, to provide a holistic understanding of academic performance. By leveraging temporal dependencies and correlations among multiple features, the model effectively captures complex learning patterns and interactions. This multidimensional approach enables more accurate predictions and offers deeper insights into the factors influencing student success.

3. Proposed Methodology

The proposed system is a web-based Student Performance Management System designed to automate and simplify academic management activities in educational institutions. This system provides a single platform for administrators, faculty, and students to perform their respective tasks efficiently. It replaces the manual method of maintaining paper records with a digital system that stores, manages, and analyses

academic data in a structured manner. In this research, the system is developed using Django as the backend framework, SQLite3 as the database, and web technologies for the user interface. The project not only manages academic records but also uses ML models such as NB, and XGB to predict student performance based on marks data. Among these models, XGB is selected for final prediction because it provides better accuracy as demonstrated in Figure 2.

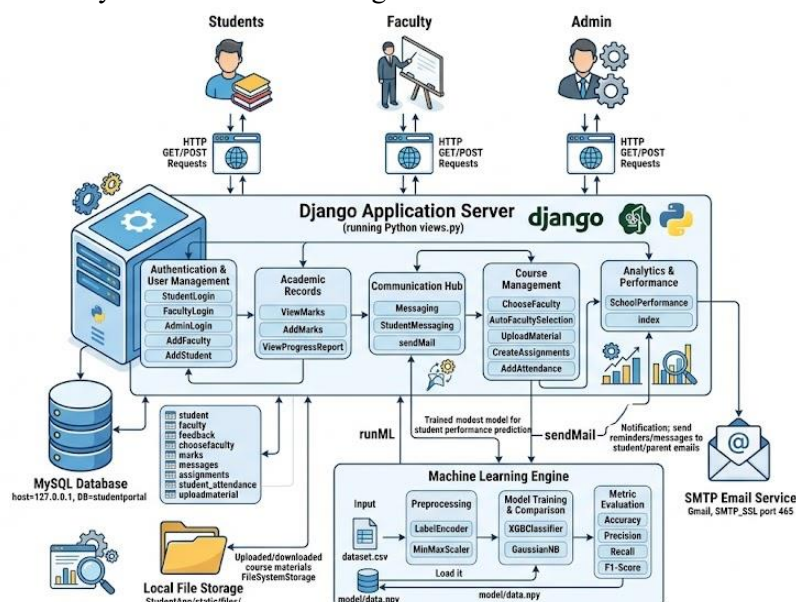


Figure. 2: Proposed system architecture

Data Collection: All academic data, including attendance, assignment submissions, test scores, faculty evaluations, and historical academic records, are collected from students and faculty. This ensures a centralized repository of student performance data for analysis.

Data Preprocessing: Raw data is cleaned, normalized, and missing values are handled. Feature selection and extraction are performed to create a structured dataset suitable for machine learning algorithms.

Centralized Database Management: The pre-processed data is stored in a centralized database, which maintains secure access and ensures consistency. This module also handles role-based access for students, faculty, and administrators.

Predictive Modelling: Machine learning models, such as Random Forest, Logistic Regression, or XGB, are trained on historical academic data to predict student performance in real-time. This module generates performance scores and identifies students who may require additional support.

Dashboard & Visualization: A dynamic dashboard presents insights from predictions, showing performance trends, at-risk students, and analytics to both faculty and students. Visualization tools enhance understanding and decision-making.

Feedback & Intervention: The system provides actionable feedback to students and faculty. Faculty can intervene early with tailored guidance, while students can track their progress and improve their academic performance proactively.

4. Implementation Description

The implementation of the intelligent student–faculty management framework is carried out through a systematic sequence of predefined and inbuilt Python and Django functions. This structured workflow ensures smooth transformation of raw academic data into meaningful insights and intelligent predictions.

1. Environment & Web Framework Initialization

The execution begins with the initialization of the Django environment and required libraries.

- `django.shortcuts.render()`: An inbuilt Django function used to render HTML templates with dynamic data.
- `django.http.HttpResponse()`: Used to send HTTP responses such as file downloads to the client.
- `FileSystemStorage()`: A predefined Django utility for handling file uploads and storage.
- `pymysql.connect()`: Establishes a connection between the Django application and the MySQL database.
- `datetime.now()`: Retrieves the current timestamp for storing activity logs such as messages, feedback, and uploads.

2. Data Ingestion & Preprocessing

- `pd.read_csv()`: Loads the dataset (dataset.csv) into a Pandas DataFrame for analysis.
- Label Encoding:
 - `LabelEncoder().fit_transform()` converts categorical feedback labels into numerical values.
- Min-Max Scaling:
 - `MinMaxScaler().fit_transform()` normalizes feature values to a uniform range.
- `np.load()`:
 - Loads pre-saved training and testing datasets for faster execution.

3. Dataset Partitioning

- `train_test_split()`:
 - Splits the dataset into training (80%) and testing (20%) subsets.
 - Ensures unbiased model evaluation by separating unseen data.

4. Model Building & Training

The system employs two machine learning algorithms:

- Naive Bayes:
 - `GaussianNB().fit()` trains a probabilistic model based on feature independence assumptions.
- XGBoost:
 - `XGBClassifier().fit()` builds an ensemble boosting model for improved prediction accuracy.
- `model.predict()`:
 - Generates predictions on the testing dataset.

5. Performance Evaluation

- `accuracy_score()`:
 - Calculates classification accuracy.
- `precision_score()`:
 - Measures prediction correctness.
- `recall_score()`:
 - Evaluates model sensitivity.
- `f1_score()`:
 - Computes harmonic mean of precision and recall.
- `confusion_matrix()`:
 - (Optional) Generates classification matrix for performance visualization.
- `sns.heatmap()` & `plt.savefig()`:
 - Used to visualize and store evaluation results.

6. Intelligent Prediction Module

- `runML()`:
 - A custom-defined function that takes student GPA as input.
 - Applies scaling using `MinMaxScaler`.

- Uses trained XGBoost model to predict feedback category.
- Returns interpreted label using stored class mappings.

7. Database Interaction Layer

- pymysql.connect():
 - Connects to the MySQL database (studentportal).
- cursor.execute():
 - Executes SQL queries for CRUD operations.
- Tables handled:
 - student, faculty, marks, feedback, messages, assignments, uploadmaterial, attendance.

8. Student Module Implementation

- View Marks:
 - Calculates total and average GPA.
 - Calls runML() for feedback prediction.
- Faculty Selection:
 - Uses getClassMembers() to limit class strength.
- Feedback Submission:
 - Stores ratings and comments in the database.
- Messaging:
 - Sends messages to faculty using database insertion.

9. Faculty Module Implementation

- Add Marks:
 - Records student performance data.
- Upload Materials:
 - Uses file handling functions to store study resources.
- Assignment Management:
 - Allows faculty to create and assign tasks.
- Attendance Tracking:
 - Records student attendance using batch insertion.

10. Admin Module Implementation

- Faculty & Student Management:
 - Add/view records using database queries.
- School Performance Analysis:
 - plt.pie() generates visualization of teacher effectiveness.
- Dashboard Rendering:
 - Displays performance metrics dynamically using templates.

11. Communication & Notification System

- smtplib.SMTP_SSL():
 - Establishes secure email communication.
- sendmail():
 - Sends notifications to students or parents.
- getEmail():
 - Retrieves email IDs from the database.

12. File Handling & Resource Management

- File Upload:
 - request.FILES[] handles uploaded content.
- File Storage:
 - Stored in static/files/ directory.
- File Download:

- `HttpResponse()` used to send files to users.

13. Visualization & Reporting

- `matplotlib.pyplot`:
 - Generates charts for performance analysis.
- `seaborn`:
 - Enhances visualization with heatmaps and plots.
- `base64.b64encode()`:
 - Converts plots into displayable web format.

4.1 Result Description

The system produced accurate and meaningful results in managing academic data as well as predicting student performance using machine learning. Blow screens are able to view the marks, attendance, assignments, study materials, and academic updates through their personalized dashboard, while faculty and administrators efficiently performed academic management tasks such as entering marks, uploading material, and reviewing student progress.

Student Performance Predictor & Tracker

Algorithm Name	Accuracy	Precision	Recall	FSCORE
Naive Bayes	86.92307692307692	94.03508771929823	81.79271708683474	85.55022771063236
XGBoost	93.07692307692308	93.66666666666667	92.5619478560655	92.84477124183006

Figure. 3: Interface overview.

Figure 3 displays the main interface of the system, providing an organized dashboard for both teachers and students. It includes navigation options for uploading data, creating assignments, managing attendance, and accessing learning materials. The interface ensures easy access to all core functionalities of the research.

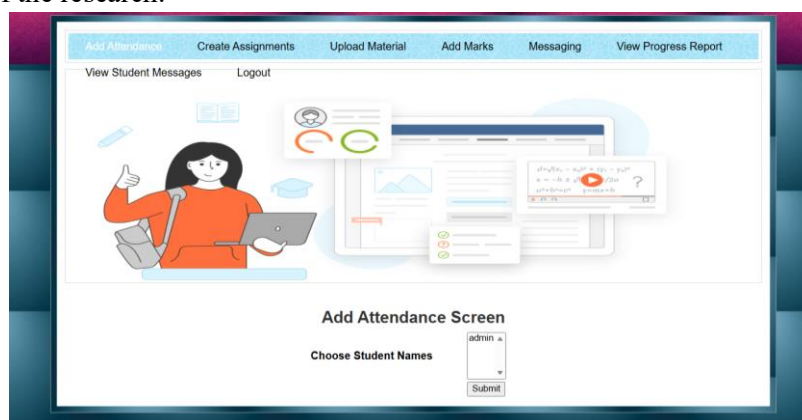


Figure. 4: Teacher uploading student attendance.

Figure 4 shows the teacher marking or uploading attendance for students. Attendance data is recorded and associated with specific dates and subjects. This feature ensures accurate tracking of student presence and supports performance analysis.

Figure 5 presents the process where teachers create assignments for students. Teachers input assignment details, including title, description, and submission deadlines. The system stores these assignments, making them accessible for students to review and submit.

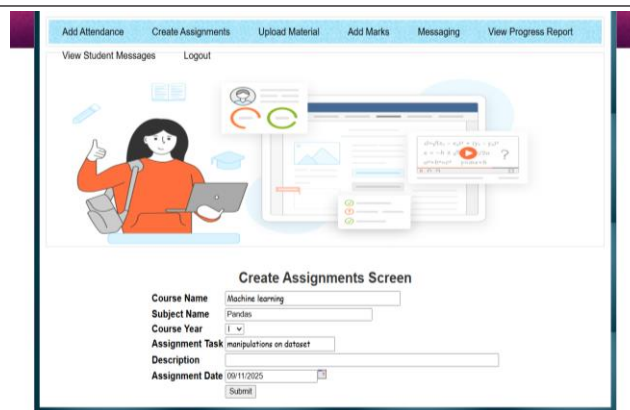


Figure. 5: Teacher creating assignments.

Table 1: Performance comparison for the NBC Classifier and XGB algorithms.

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
NBC	86.92	94.03	81.79	80.55
XGB	93.07	93.66	93.56	92.84

The Table 1 presents a performance comparison between the NBC Classifier and the XGB algorithm. The NBC Classifier achieved an accuracy of 86.92%, with precision, recall, and F1-score values of 94.03%, 81.79%, and 80.55%, respectively. In contrast, the XGB Classifier outperformed NBC across all metrics, achieving an accuracy of 93.07%, precision of 93.66%, recall of 93.56%, and an F1-score of 92.84%. This comparison highlights the stronger predictive capability of XGB in handling the dataset used for this project. The results demonstrate that XGB provides more balanced performance between precision and recall, ensuring higher reliability in classification tasks.

Table 2 shows the performance metrics of the NBC Classifier on the dataset. The model achieved an accuracy of 86.92%, indicating a strong overall correctness in predictions. Precision reached 94.03%, reflecting a high proportion of correctly identified positive instances. Recall was 81.79%, showing the model's ability to identify relevant cases, while the F1-score of 80.55% represents a balanced measure of precision and recall.

Table 2 Performance metrics of NBC Classifier.

Metric	Value
Accuracy	86.92
Precision	94.03
Recall	81.79
F1-Score	80.55

Table 3 presents the performance metrics of the XGB Classifier on the dataset. The model achieved an accuracy of 93.07%, demonstrating excellent overall prediction capability. Precision was 93.66%, indicating a high rate of correctly predicted positive instances. Recall reached 93.56%, reflecting the model's effectiveness in identifying relevant cases. The F1-score of 92.84% highlights a strong balance between precision and recall, confirming the robustness of the XGB model.

Table.3 Performance metrics of XGB Classifier.

Metric	Value
Accuracy	93.07
Precision	93.66
Recall	93.56
F1-Score	92.84

5. Conclusion

The research successfully developed an automated educational management system integrating teacher and student functionalities. Teachers were able to upload student details, mark attendance, and create assignments efficiently, while students could access assignments and download study materials seamlessly. The implementation of machine learning models, specifically NBC and XGB, enhanced the system's predictive and analytical capabilities. XGB delivered superior performance with an accuracy of 93.07%, high precision, recall, and F1-score, proving the model's effectiveness in handling educational data. The system improved data management, reduced manual workload, and provided a structured, efficient interface for both teachers and students.

References

- [1] Kezar, A.; Kitchen, J.A.; Estes, H.; Hallett, R.; Perez, R. Tailoring programs to best support low-income, first-generation, and racially minoritized college student success. *J. Coll. Stud. Retent. Res. Theory Pract.* 2023, 25, 126–152.
- [2] Wilmer, E. Student support services for the underprepared student. *Inq. J. Va. Community Coll.* 2008, 13, 4–19.
- [3] Andrews, R.J.; Imberman, S.A.; Lovenheim, M.F. Recruiting and supporting low-income, high-achieving students at flagship universities. *Econ. Educ. Rev.* 2020, 74, 101923.
- [4] Hoyt, J.E. Student connections: The critical role of student affairs and academic support services in retention efforts. *J. Coll. Stud. Retent. Res. Theory Pract.* 2023, 25, 480–491.
- [5] Kuh, G.D. Student success. In *Student Services: A Handbook for the Profession*, 5th ed.; Schuh, J.H., Jones, S.R., Harper, S.R., Eds.; Jossey-Bass: Hoboken, NJ, USA, 2011; pp. 257–270.
- [6] Alhothali, A.; Albsisi, M.; Assalahi, H.; Aldosemani, T. Predicting Student Outcomes in Online Courses Using Machine Learning Techniques: A Review. *Sustainability* 2022, 14, 6199. <https://doi.org/10.3390/su14106199>.
- [7] Alruwais, N.; Zakariah, M. Evaluating Student Knowledge Assessment Using Machine Learning Techniques. *Sustainability* 2023, 15, 6229. <https://doi.org/10.3390/su15076229>.
- [8] Adefemi, K.O.; Mutanga, M.B.; Jugoo, V. Hybrid Deep Learning Models for Predicting Student Academic Performance. *Math. Comput. Appl.* 2025, 30, 59. <https://doi.org/10.3390/mca30030059>.
- [9] Namoun, A.; Alshanqiti, A. Predicting Student Performance Using Data Mining and Learning Analytics Techniques: A Systematic Literature Review. *Appl. Sci.* 2021, 11, 237. <https://doi.org/10.3390/app11010237>.
- [10] Jawad, K.; Shah, M.A.; Tahir, M. Students' Academic Performance and Engagement Prediction in a Virtual Learning Environment Using Random Forest with Data Balancing. *Sustainability* 2022, 14, 14795. <https://doi.org/10.3390/su142214795>.
- [11] Baashar, Y.; Alkawsi, G.; Mustafa, A.; Alkahtani, A.A.; Alsariera, Y.A.; Ali, A.Q.; Hashim, W.; Tiong, S.K. Toward Predicting Student's Academic Performance Using Artificial Neural Networks (ANNs). *Appl. Sci.* 2022, 12, 1289. <https://doi.org/10.3390/app12031289>.
- [12] Mohammad, A.S.; Al-Kaltakchi, M.T.S.; Alshehabi Al-Ani, J.; Chambers, J.A. Comprehensive Evaluations of Student Performance Estimation via Machine Learning. *Mathematics* 2023, 11, 3153. <https://doi.org/10.3390/math11143153>.
- [13] Shou, Z.; Xie, M.; Mo, J.; Zhang, H. Predicting Student Performance in Online Learning: A Multidimensional Time-Series Data Analysis Approach. *Appl. Sci.* 2024, 14, 2522. <https://doi.org/10.3390/app14062522>.